# 操作系统实验日志

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 201808010718 | 姓名 | 肖鹏 | 专业年级班级 | 智能1802 |
| 实验日期 | 2020.9.24 | 实验项目 | 第1天：从计算机结构到汇编程序入门  第2天：汇编语言学习与Makefile入门 | | |

## 一、实验主要内容

1、

**内容：**如何生成自己的映像文件、以及通过命令行运行自己的简易“操作系统”；

**预处理：**下载“二进制编辑器”——Bz.exe；

**重点总结：**

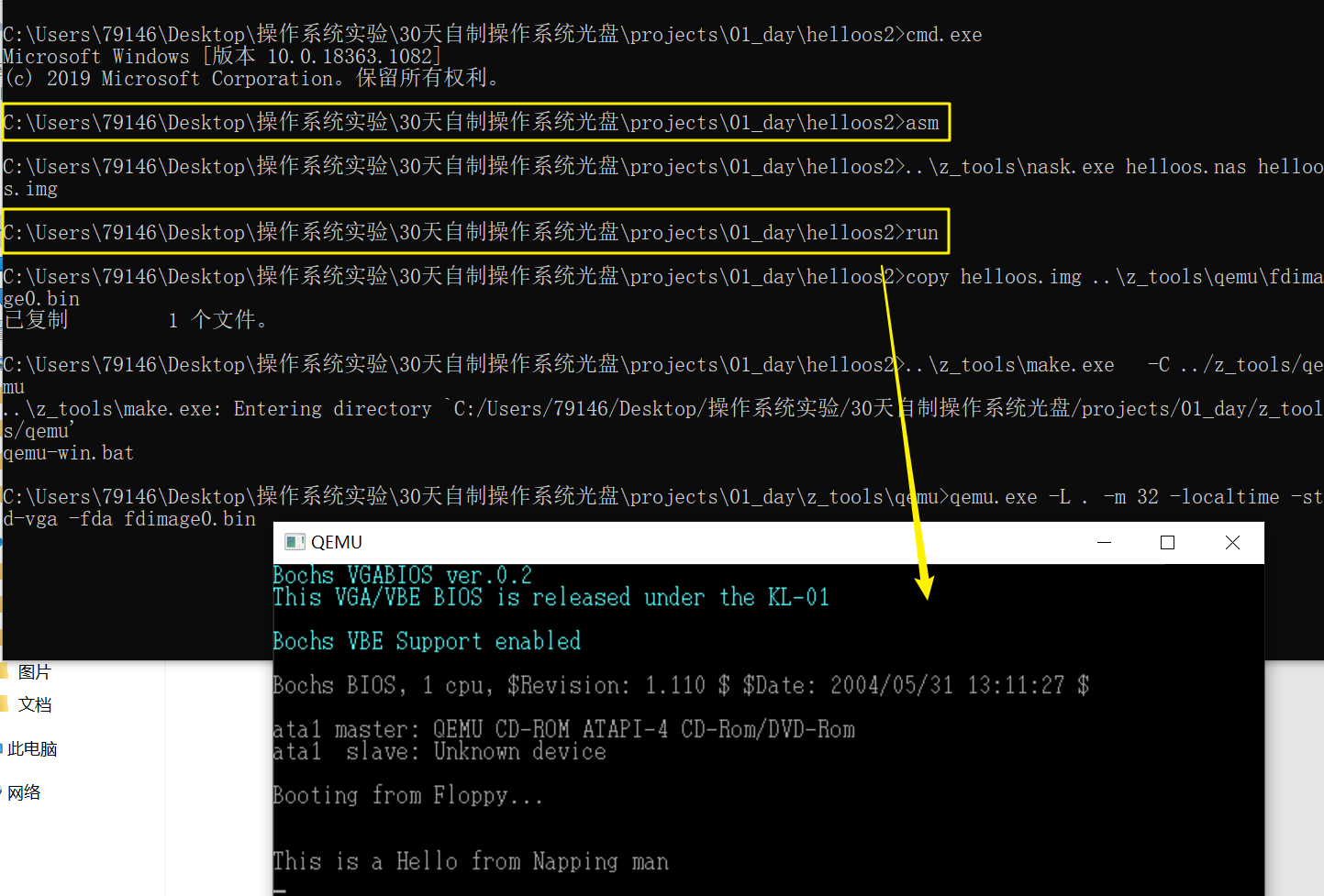
①首先通过Bz.exe写入机器码，保存为映像文件helloos.img。

②install.bat文件是用来把helloos.img装入软盘的程序指令，然而我们是直接在Windows机器中操作，所以并没有实际用到install.bat。

③run.bat文件中汇总了几条用来运行映像文件的指令，用到了z\_tools中的程序，所以z\_tools文件夹应该被复制到tolset中。

④ams.bat文件中指令作用是将helloos.nas中的机器码或汇编代码，通过程序nask.exe生成映像文件。

**关键代码及注释：**



2、

**内容：**批处理文件理入门 ；

**重点总结：**

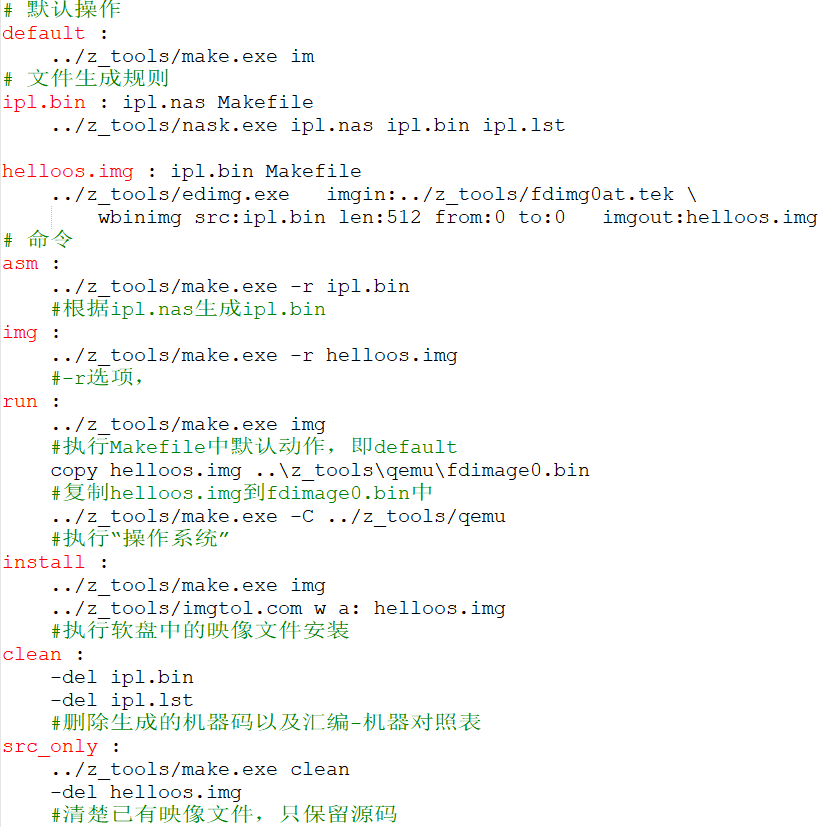
通过Makefile文件，把之前单独使用的asm.bat、run.bat、install.bat集中，简化操作指令；并且增加删除功能。

其中Makefile的一些语法总结：①冒号之后是依赖文件，即执行该指令时冒号后列出的文件必须存在；

命令部分就和普通的在命令行中意义的一样。

**关键代码及注释：**

Makefile文件及注释



3、

## 二、遇到的问题及解决方法

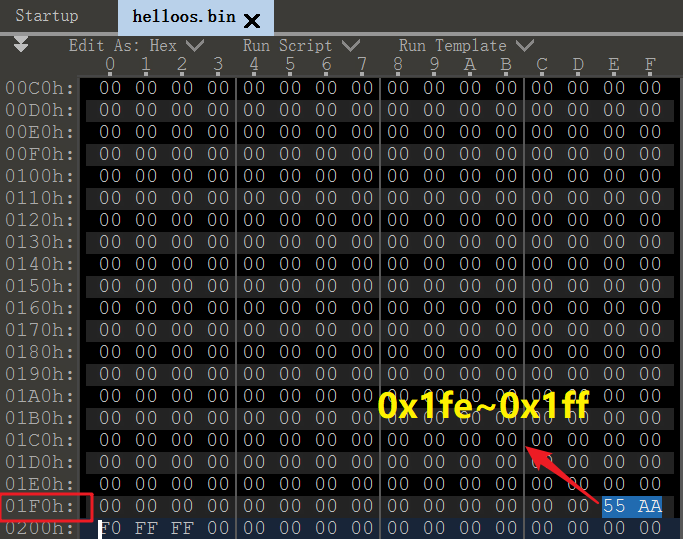
1、

**问题：**对启动区RESB 0x1fe-$的理解。

**导致原因：**对于启动区代码规模的陌生，以及DB、DW、DD指令的认识偏差，这些指令的本质意义是将指令后的操作数无差别的视作1个byte、1个word、两个word，而DB指令写字符串只是另外一种它的用法。对指令作用的误解直接导致对启动区代码占用字节数计算的不理解，以及对教材中给出的’$’等于132产生困惑。

**解决方法：**在理解了DB、DW、DD指令的实际作用后，重新分析了启动区代码占用的字节数，发现：FAT12格式软盘专用代码占据80 bytes、程序主题占据36 bytes、（RESB指令之前的）信息显示部分占据16 bytes，总计80+36+16=132 bytes，这个数据也就是教材中说的’$就是132’。在教材有关启动区的概念中，指出在第一个扇区，也就是启动区的最后两个字节需为55 AA，才能标即这个扇区是启动区，否则将抛出异常，通过二进制编辑器查看helloos0中的源码（如下图），可以验证这一点。

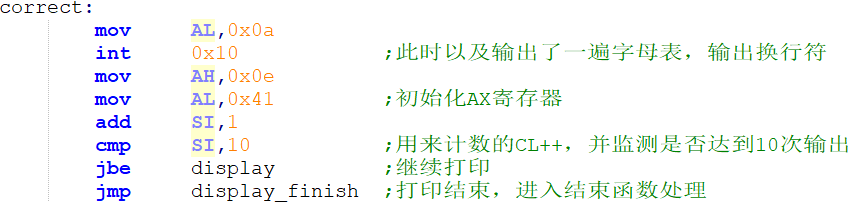
一个扇区512 bytes，即0x200 bytes，最后两个bytes地址为0x1fe、01ff。



2、

**问题：**在实现“创新点2”时，其中有一个设计是控制循环输出字母表的次数，需要用到寄存器来计数。由于在教材中看到有用SI寄存器来保留途中计数值并且用来边际比较的代码，所以一开始也用的是寄存器SI，但是发现在helloos5中的ipl.nas代码中，使用该寄存器会出现异常的行为：比如设定循环次数为5，在correct函数中，每进入一次correct，SI++，并且进行边界判断，通过跳转执行不同行为；

（附上导致错误的代码和现象）

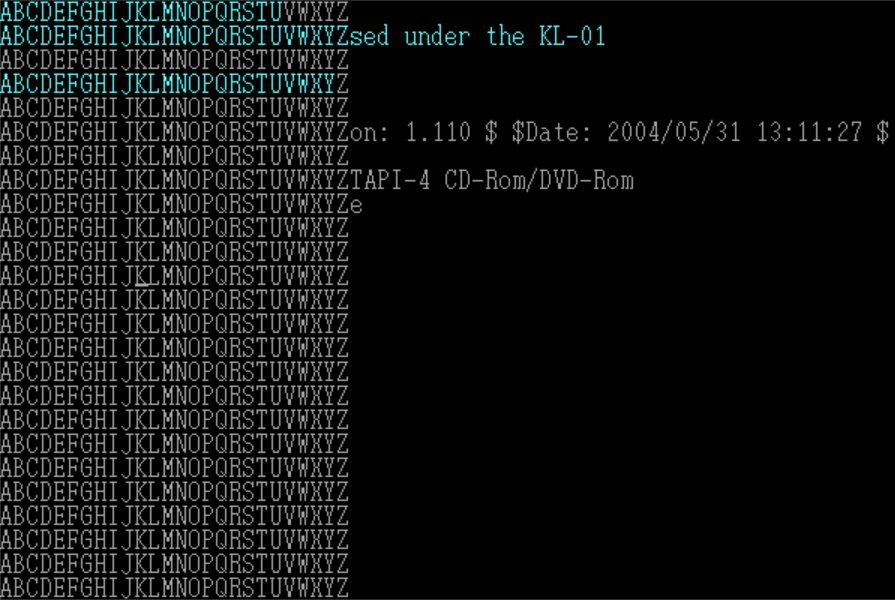
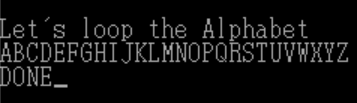


**异常现象：**

如图所示，当使用SI寄存器时：

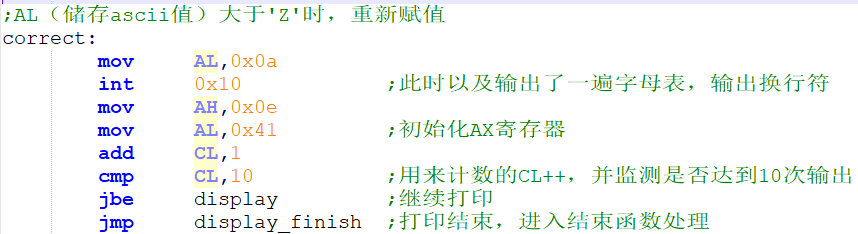
①若jbe跳转到display继续打印，则只会打印一次便提示“DONE”；

②若jae跳转到display继续打印，则将陷入无限的字母表打印中，且在最后一行输出之后会回到第一行覆盖原有的信息。



**导致原因：**初步认为是SI寄存器在ipl.nas文件中其他代码中使用切改变了SI的值；所以SI的值并不是按预期的变化，导致观察到的跳转异常。

**解决方法：**使用计数寄存器CL替代SI，CL在代码中没有被使用过。



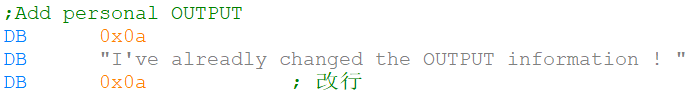
## 三、程序设计创新点

1、

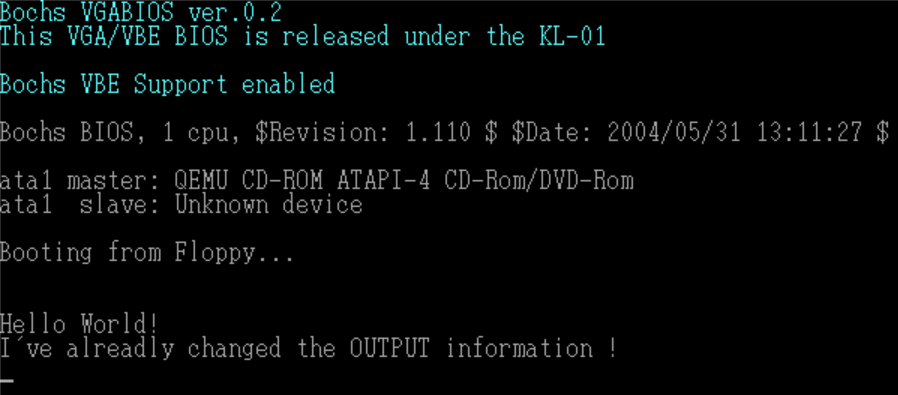
**创新点：**

修改初始界面“Hello World！”字串。通过观察ipl.nas中的指令，DB “string”，可以输出字符串，修改是的输出自己的字符串。

**关键代码：**



结果截图：

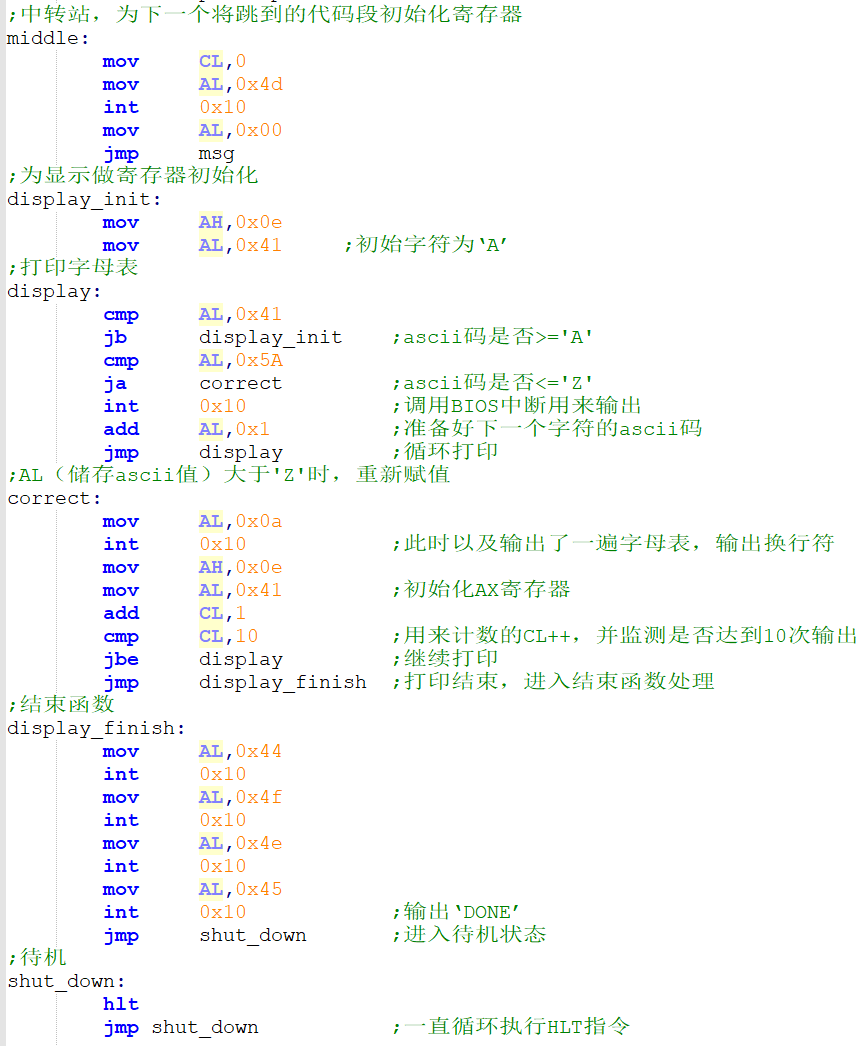


2、

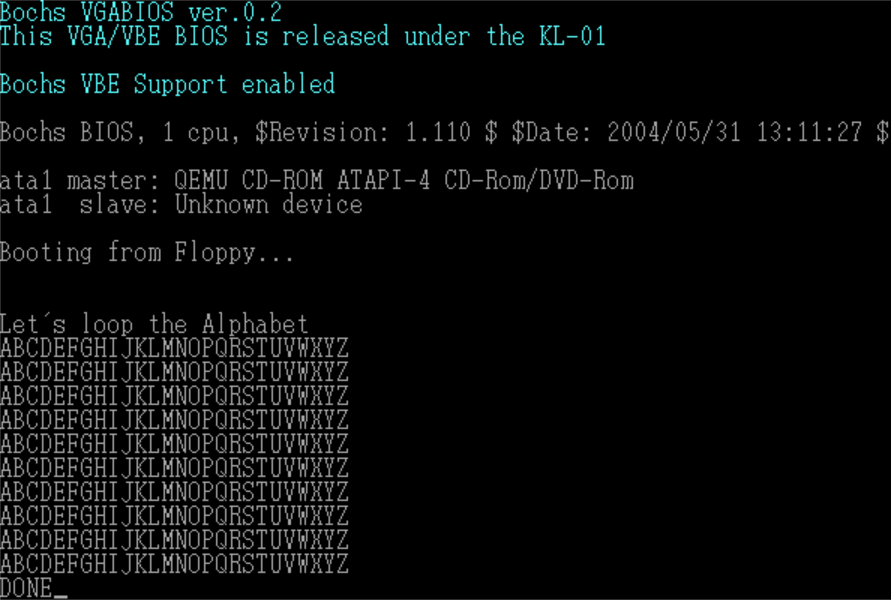
**创新点：**

通过修改、编写汇编代码，使得循环输出“A~Z”字母表。通过这个测试熟悉了通过中断INT 0x10，把相应字符的ASCII码装入AL寄存器以实现print功能；之后，通过寄存器CL，实现累加与判断边界功能，从而通过以不同标志位为跳转依据的JMP指令，使得函数之间实现互相调用，或者借助HLT指令，实现待机。最终实现循环输出10遍字母表后输出提示符“DONE”后进入待机状态。

**关键代码：**



**结果截图：**



## 四、实验心得体会

这次实验算是这门实验课的入门，第1、2天的实验主要是熟悉了具体的操作——通过执行几条指令，从源码到映像文件再到最后的执行结果，如何观察自己的“操作系统”是否按预期执行。然后对一些基本的指令进行解释、Makefile的语法有了初步的认识，主要的是对汇编语言编程有了更进一步的体会，这是第一次通过自己写intel汇编代码实现利用中断输出字符、通过JMP指令在不同函数之间的调用。 由于比较陌生，在调试的时候只能通过注释法，把增加的代码一段一段的注释，最后找到表现异常的代码段进行调试。